

CLAIMS

What is claimed is:

- 1 1. A method, comprising:
 - 2 dynamically establishing a first debugging session with a first processing
 - 3 core of a processor;
 - 4 dynamically establishing a second debugging session with a second
 - 5 processing core of the same processor; and
 - 6 concurrently managing the first and second debugging sessions
 - 7 independently from one another on the same processor.

- 1 2. The method of claim 1, further comprising:
 - 2 initiating a first instance of a debugger on the first processing core of the
 - 3 processor for interactions occurring during the first debugging session; and
 - 4 initiating a second instance of the debugger on the second processing core of
 - 5 the processor for interactions occurring during the second debugging session.

- 1 3. The method of claim 1, wherein dynamically establishing the first and
- 2 second debugging sessions further comprises dynamically establishing the sessions
- 3 by connecting the first and second processing cores of the processor to separate
- 4 instances of a debugger via a Peripheral Component Interconnect (PCI) interface.

- 1 4. The method of claim 1, further comprising:
 - 2 debugging a first application within the first debugging session on the first
 - 3 processing core of the processor; and
 - 4 simultaneously debugging a second application within the second debugging
 - 5 session on the second processing core of the processor.

1 5. The method of claim 1, further comprising:
2 identifying within a first configuration file of a first debugger the first
3 processing core associated with the first debugging session; and
4 identifying within a second configuration file of a second debugger the
5 second processing core associated with the second debugging session.

1 6. The method of claim 5, further comprising:
2 routing, by the processor, the first debugger to the first processing core for
3 establishing the first debugging session based on the first configuration file; and
4 routing, by the processor, the second debugger to the second processing core
5 for establishing the second debugging session based on the second configuration
6 file.

1 7. The method of claim 1, further comprising maintaining, by the processor,
2 processor states while dynamically establishing the first and second debugging
3 session.

1 8. A method, comprising:
2 receiving, by a processor, a first debugging session request;
3 receiving, by the processor, a second debugging session request;
4 dynamically attaching a first debugger to a first processing core for servicing
5 the first debugging session request; and
6 dynamically attaching a second debugger to a second processing core for
7 servicing the second debugging request.

1 9. The method of claim 8, wherein dynamically attaching the first and second
2 debuggers further includes identifying the first and second debuggers as a same
3 debugger being initiated as independent and duplicative instances on different
4 processing cores.

1 10. The method of claim 8, further comprising identifying within the first and
2 the second debugging session requests configuration information which identifies
3 the first and second processing cores.

1 11. The method of claim 8, wherein dynamically attaching the first and second
2 debuggers further includes maintaining a previous state associated with the
3 processor of the first and second processing cores before and after attaching the first
4 and second debuggers to their respective processing cores.

1 12. The method of claim 8, wherein receiving the first and second debugging
2 session requests further includes remotely initiating the requests from the processor
3 that has the first and second processing cores.

1 13. The method of claim 8, further comprising maintaining existing states
2 associated with existing applications, the existing applications processing on the
3 first and second processing cores before and after dynamically attaching the first
4 and second debuggers to the first and second processing cores, respectively.

1 14. The method of claim 8, wherein dynamically attaching the first and second
2 debuggers further includes attaching the first and second debuggers to their
3 respective processing cores as their respective processing cores are processing a
4 number of other applications.

1 15. A system, comprising:
2 a processor having a first processing core and a second processing core; and
3 a debugger, wherein a first instance of the debugger is dynamically
4 attachable to the first processing core of the processor and a second instance of the
5 debugger is dynamically attachable to the second processing core of the processor.

1 16. The system of claim 15, further comprising a Peripheral Component
2 Interconnect (PCI) interfaced to the processor for receiving requests to dynamically
3 attach the first and second debugger instances to their respective processing cores.

1 17. The system of claim 15, further comprising a first configuration file
2 associated with the first debugging instance and a second configuration file
3 associated with the second debugging instance, wherein each configuration file
4 identifies its respective processing core, and wherein the processor in response to
5 the configuration files dynamically attaches the debugger instances to their
6 respective processing cores.

1 18. The system of claim 15, wherein the first debugging instance establishes a
2 first debugging session for debugging a first application and the second debugging
3 instance establishes a second debugging session for debugging a second application,
4 and wherein the first and second applications are different from one another.

1 19. The system of claim 15, wherein the processor maintains states associated
2 with the first and second processing cores before and after the first and second
3 instances are dynamically attached to their respective processing cores.

1 20. A machine accessible medium having associated instructions, which when
2 accessed, results in a machine performing:
3 receiving a first request for a first debugging session;
4 receiving a second request for a second debugging session;
5 dynamically establishing the first debugging session on a first processing
6 core; and
7 dynamically establishing the second debugging session on a second
8 processing core.

1 21. The medium of claim 20, further including instructions for concurrently
2 managing the first and second debugging sessions independent from one another on
3 a processor having the first and second processing cores.

1 22. The medium of claim 20, further including instructions for initiating a first
2 debugger instance on the first processing core for managing the first debugging
3 session and a second debugger instance on the second processing core for managing
4 the second debugging session.

1 23. The medium of claim 20, further including instructions for maintaining
2 states of the first and second processing cores before and after the first and second
3 debugging sessions are dynamically established on their respective processing cores.

1 24. The medium of claim 20, further including instructions for actively and
2 simultaneously debugging a first application within the first debugging session and
3 a different application within the second debugging session.

1 25. An apparatus, comprising:
2 configuration information associated with instances of a debugger; and
3 attachment logic residing within a processor having multiple processing
4 cores that dynamically attaches selective ones of the instances of the debuggers to
5 selective ones of the processing cores in response to requests for debugging sessions
6 having the configuration information.

1 26. The apparatus of claim 25, wherein the attachment logic maintains states of
2 the processing cores before and after any dynamic attachment of the instances of the
3 debugger.

1 27. The apparatus of claim 25, wherein configuration information is
2 configurable parameter values provided with the requests for debugging sessions.

- 1 28. The apparatus of claim 25, wherein the configuration information is files,
- 2 where each file is associated with a unique one of the debugging instances.